# Bridging Real-to-Sim Gaps through Online Stiffness Optimization with Perception-Enabled Residual Mapping

Xiao Liang[†], Fei Liu[†], Yutong Zhang, Michael Yip

***Abstract*— Accurate deformable object manipulation (DOM) is an essential component for achieving autonomy in robotic surgery, which involves operations on soft tissue. Many DOM algorithms can be powered by simulation, which ensures realistic deformation by adhering to the governing physical constraints. However, real soft objects in robotic surgery have complex, anisotropic physical parameters and topology that a simulation with simple initialization cannot capture. To use the simulation technique in real surgical tasks, the "real-to-sim" gap needs to be properly compensated. In this work, we propose to use an optimization-based residual mapping module to close the positional gap between a physics simulation and perceptual observation. We further use this module to guide an adaptive online estimation of the initialized physics parameters of soft bodies using a position-based dynamics (PBD) simulator. The proposed method is able to produce more realistic soft body deformation. The performance of the proposed mechanism is evaluated in the manipulation of a real thin-shell tissue manipulated by the Da Vinci Surgical System.**

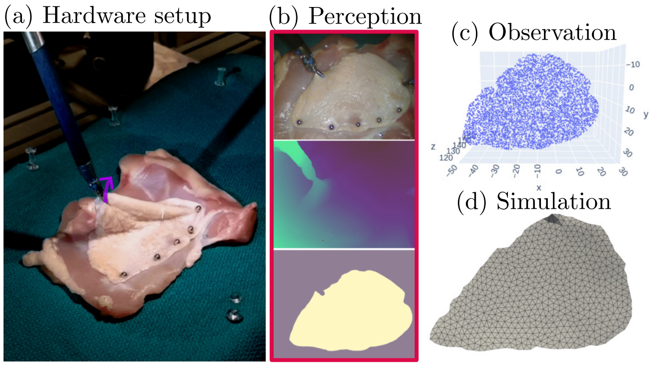

Fig. 1. The real-to-sim experiment setup in this work. (a): a piece of chicken muscle manipulated by a dVRK manipulator. (b): A perception pipeline estimates depth and a semantic mask of the tissue. (c): surface point cloud is generated by a camera inverse projection. (d): A simulation mesh is created with the initial observation.

## I. INTRODUCTION

The physics-based simulation has been proven to be a promising technique for deformable object manipulation (DOM) in robotics [1], [2], [3], in particular for surgical applications [4], [5]. It has been extensively researched from the modeling, motion planning, and data representations from perception perspectives. The development of physical simulation environments can support a variety of deformable objects, including thin-shell fabric, linear elastic ropes [6], and volumetric tissue [7].

Considering many existed deformable simulators, the high computational cost and "reality gap" are two factors that limits the conventional simulation approaches, such as FEM, mass-spring system [8], [1]. Uncertainties, inaccurately calibrated parameters, and unmodeled physical effects can all lead to the gap. As a result, many simulators need their parameters fine-tuned before being depolyed into actual robotic systems. With the intention of bridging the gap between reality and simulation, the phrase "real-to-sim" is used to describe approaches that compensate the simulated errors from real observations.

Therefore, our goal is to investigate the "real-to-sim" gaps by taking into account both physical constraints and observable high-dimensional data, i.e., point cloud. For real-time applications such as surgical tissue grasping, it would also be necessary to identify the proper simulation parameters and the morphology of soft bodies with a fast online simulation approach.

### *A. Related Works*

The "real-to-sim" problem have gained attention recently in literature. Most of the existed works have been focusing on closing the gap through effective policy transfer in a reinforcement learning setup, as reviewed in [9], [10]. Most of these works only rely on simulating rigid objects in the scene and robots with kinematics [11]. Recently, several papers use similar deep reinforcement learning for deformable objects, such as cloth [12], tissue [13], ropes [14]. However, these works don't create explicit physical models and instead learning the system parameters or controls in an end-to-end manner. It leads to several typical problems with learning strategies, including generalizability, accuracy and effectiveness.

Other methods rely on physics-based simulation to minimize the "real-to-sim" gap. In our previous paper [7], we directly update the simulated positions of the volumetric particles using the spatial gradient of the signed distance field of point cloud observation. However, the simulation parameters are not updated, necessitating frame-by-frame registration. Similarly, [15] optimizes for a simulation parameter, such as mass or stiffness, to minimize the difference between a simulation and the point cloud observation. Furthermore, the authors enhanced their approach in [16] as probabilistic inference over simulation parameters of the deformable object. However, their method is limited to thin-shell or linear objects (cloth, rope) with surface point clouds. Projective dynamics [**?**] were used to create a real-

[†] Equal contributions.

[1]Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093 USA.{x5liang, f4liu, yuz049, yul189, yip}@ucsd.edu

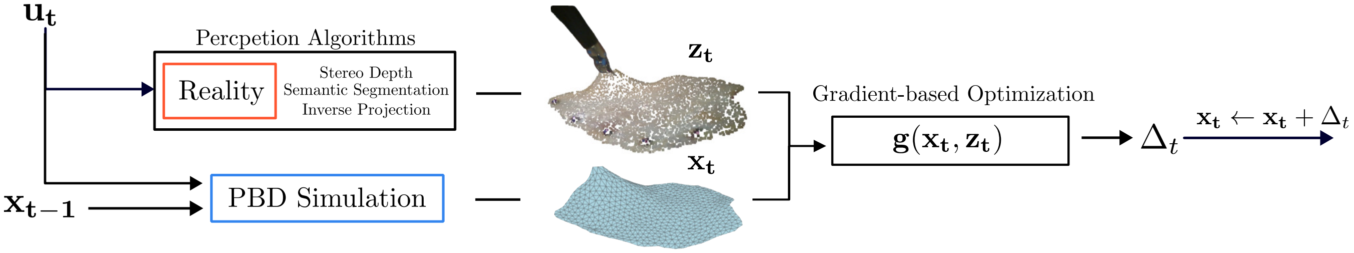

Fig. 2. A flow chart of the proposed residual mapping module in the simulation loop. At each time step, a control $\mathbf{u_t}$ is applied to both the real tissue and the simulation. In response, the PBD simulation solves for $\mathbf{x_t}$. A perception pipeline processes imagery data to obtain a surface point cloud $\mathbf{z_t}$ of the tissue. The residual mapping module estimates the residual deformation $\Delta_t$ via optimization, which is then used to update the simulation state.

time physics-based model for tissue deformation, enhanced by a Kalman filter (KF) for refining the simulation with surface marker data. For further improvements, [17] utilized a variational autoencoder with graph-neural networks to learn low-dimensional latent state variables' probability distributions. These variables were iteratively updated using an ensemble smoother with data assimilation to align the simulation with real data. However, their offline training for specific FEM simulation datasets poses challenges in real-world applications, especially in surgery, where lengthy data-collection and pretraining phases are impractical.

### B. *Contributions*

In this work, we propose a residual mapping module to compensate for the real-to-sim gap between actual observation and a simulation. The proposed framework can reduce the positional gap while maintaining plausible geometric and physical behavior. To this end, we present the following novel contributions:

- To compensate for the real-to-sim gap, we propose an optimization-based residual mapping module to accurately match simulation states to real observation while respecting geometric constraints;
- We are the first online method to incorporate the residual mapping module into the physical simulation loop. By updating the simulation parameters online, our method adaptively reduces the real-to-sim gap.
- We set up a perception pipeline and conducted a real thin-shell tissue manipulation experiment to evaluate the proposed method.

## II. METHOD

### A. *Problem Formulation*

Let the state of a soft body in a physics simulation at time stamps $t$ be $\mathbf{x_t} \in \mathbb{R}^{n\times 3}$, where $n$ is the number of particles of a mesh representing the soft body. We use the surface point cloud projected from the stereo-depth estimation as the observation, denoted as $\mathbf{z_t} \in \mathbb{R}^{m\times 3}$, where $m$ number of points in the point cloud observation. Let $\mathbf{u}_t$ be a point-based positional control that is applied to the real object and simulation simultaneously. In this work, we use an extended position-based dynamic (PBD) simulator, which formulates constraints with positional and geometric data. A constraints-based formulation of PBD is given as

$$\begin{aligned} \mathbf{x_t} &= \mathcal{PBD}(\mathbf{x_{t-1}}, \mathbf{u_t}, \boldsymbol{\Gamma}, \mathbf{C}, \mathbf{k_c}) \\ &s.t., \mathbf{C}(\mathbf{x_t}) = \mathbf{0},\ \boldsymbol{\Gamma}(\mathbf{x_t}) = \mathbf{0} \end{aligned} \qquad (1)$$

where $\boldsymbol{\Gamma}$ are static (boundary) conditions that are enforced at each simulation step. The set of geometric constraints that define the deformation of across the simulation is $\mathbf{C}(\mathbf{x}) = [C_1(\mathbf{x}), C_2(\mathbf{x}), \cdots, C_I(\mathbf{x})]^\top$. $\mathbf{k_c} \in \mathbb{R}^I$ is the set of weighting parameters associated with any kind of non-boundary constraint. One can interpret that $\{\mathbf{C}, \mathbf{k_c}\}$ jointly defines the total energy potential (unitless) generated by constraints in a simulation:

$$\mathbf{U}(\mathbf{x}) = {}^1\!/_2\ \mathbf{C}(\mathbf{x})^\top \mathtt{diag}(\mathbf{k_c})\mathbf{C}(\mathbf{x}) \qquad (2)$$

This is used by PBD to iteratively update particle positions, by minimizing this energy term:

$$\begin{aligned} \Delta\mathbf{x} &= \mathbf{M}^{-1}\nabla\mathbf{C}^\top\Delta\lambda \\ \Delta\lambda &= -\left(\nabla\mathbf{C}\mathbf{M}^{-1}\nabla^\top\mathbf{C} + \tilde{\alpha}\right)^{-1}\left(\mathbf{C} + \tilde{\alpha}\lambda\right) \\ \tilde{\alpha} &= \mathtt{diag}(\mathbf{k_c})^{-1}/\Delta t^2) \end{aligned} \qquad (3)$$

in which $\mathbf{M}$ is a diagonal mass matrix and $\lambda$ is a Lagrange multiplier vector.

In this work, we aim at optimizing stiffness parameters associated with each particle, $\mathbf{k} \in \mathbb{R}^n$. This converts to elastic constraints' weights in the simulation, denoted by $\mathbf{k_d} \subset \mathbf{k_c}$, by averaging stiffnesses across all involved particles. The value of $k_{d,i} \in \mathbf{k_d}$, that is, the weights of an elastic constraint $C_i(\mathbf{x}) \in \mathbf{C}$, is

$$k_{d,i} = {}^1\!\big/_{\mathtt{card}(Q_i)}\ \sum\nolimits_{q\in Q_i} k^q,\ \ k^q \subset \mathbf{k}$$

where particle indices $Q_i$ are considered by $C_i(\mathbf{x})$. Here, $Q_i$ contains $\mathtt{card}(Q_i)$ number of particles that are connected via mesh edges, triangles, and tetrahedrons. This achieves non-homogeneous elastic stiffness across different regions of a soft body. We apply the developed framework on thin-shell deformable bodies that a simulated with a single layer particles similar to cloth.

### B. *Real-To-Sim Residual Mapping*

In our framework, an optimization approach is used to quantify the gap between simulation and reality as shown in Figure 2. Specifically, we develop a gradient-based non-rigid point cloud registration method to estimate the residual deformation $\Delta$ from the simulation state to match the surface point cloud observation. To make the predicted residual mapping accurate and physically realistic at the same time, we consider both point cloud similarity and physical realness as cost functions to minimize.

Because the correspondence between the simulation state and point cloud observation is unknown, we use Chamfer Distance as a measurement of the similarity between two point clouds, which is computed by summing the squared distances between the nearest neighbor of two point clouds. It is defined as

$$\mathcal{D}(\mathbf{x_t}, \mathbf{z_t}) = \sum_{x \in \mathbf{x_t}} \min_{z \in \mathbf{z_t}} \|x - z\|_2^2 + \sum_{z \in \mathbf{z_t}} \min_{x \in \mathbf{x_t}} \|z - x\|_2^2 \tag{4}$$

Although Chamfer Distance alone can already match two point clouds, it potentially violates the original geometric relationship between simulation states. Previously, many techniques, such as an As-Rigid-As-Possible regularization term [18], can be used to address this problem. Instead, we can achieve the same goal by directly utilizing the geometric constraints that the PBD simulation defines. We regularize the elastic potential term:

$$\mathcal{E}(\mathbf{x_t}) = {}^1\!/_2\ \mathbf{C}(\mathbf{x_t})^\top \texttt{diag}(\mathbf{k}'_\mathbf{c})\mathbf{C}(\mathbf{x_t}) \tag{5}$$

where $\mathbf{k}'_\mathbf{c}$ is a uniform user-defined stiffness matrix. Finally, we compute a residual mapping $\Delta_t$ via minimizing

$$\mathbf{g}(\mathbf{x_t}, \mathbf{z_t}) = \min_{\Delta_t} \mathcal{D}(\mathbf{x_t} + \Delta_t, \mathbf{z_t}) + \mathcal{E}(\mathbf{x_t} + \Delta_t) \tag{6}$$

The above minimization problem is solved by performing gradient descent with a fixed step size.

### C. *Online Stiffness Optimization*

The proposed online optimization method differs from previous real-to-sim methods as it does not rely on training on previously collected trajectories. In this way, we are solving an *online* problem that is much more generalizable to real-world, unstructured environments. The algorithm is also embedded in the simulation loop as shown in Algorithm 1.

One term we want to minimize directly is the residual gap, which characterizes how much the simulation deviates from the observation for the current time step.

$$\mathcal{L}_{gap} = \|\mathbf{g}(\mathbf{x_t}, \mathbf{z_t})\| \tag{7}$$

The residual gap alone doesn't consider historical information and, therefore, is sensitive to current observation noise. To address that, we introduce a history term computed over a set of previous time points $H_t$. It is defined as

$$\mathcal{L}_{hist} = \sum\nolimits_{h \in H_t} \|\mathbf{x}_h + \Delta_h - \mathcal{PBD}(\mathbf{x}_h + \Delta_h, \mathbf{0}, \mathbf{C}, \mathbf{k_c})\| \tag{8}$$

where $\mathbf{x}_h$ and $\Delta_h$ are snapshots of simulation states and residual mapping at time point $h$. Specifically, $H_t$ is constructed by uniformly sampling four snapshots from a window of the closest 20 previous frames. This loss function finds the stiffness parameters that keep each snapshot at rest when no control is provided (i.e., balancing external forces and internal elastic forces).

In addition, we encourage a smooth spatial stiffness distribution by penalizing $\mathbf{k}$'s differences between neighboring particles. Let $F$ be a set of all faces or tetrahedrons that sorts tuples of particle indices, the smoothness loss is written as

$$\mathcal{L}_{smooth} = \frac{1}{2F} \sum_{f \in F} \sum_{i \in f} \sum_{j \in f} k_i - k_j \tag{9}$$

where $k_i \in \mathbf{k}$ is the stiffness value of the *i*-th simulation particle. Lastly, all terms are summed up and back-propagated to the stiffness parameters. They are updated by taking a stochastic gradient descent step in every simulation step.

**Algorithm 1:** Residual Mapping and Online Stiffness Optimization in a PBD simulation

```
Input : Predefined control sequence $\mathbf{U}$, stiffness $\mathbf{k}$, real tissue $R$,
        observation model $\mathcal{H}(R)$, residual mapping module $\mathbf{g}$.
$\mathbf{z_0} \leftarrow \mathcal{H}(R)$
// Initialize a mesh and constraints.
$\mathbf{x_0}, \mathbf{C} \leftarrow \texttt{initializeSimulation}(\mathbf{z_0})$
$H_t \leftarrow [\,]$
for each $\mathbf{u_t} \in \mathbf{U}$ do
    $R \leftarrow ApplyControl(R, \mathbf{u_t})$
    $\mathbf{x_t} \leftarrow \mathcal{PBD}(\mathbf{x_{t-1}}, \mathbf{u_t}, \mathbf{C}, \mathbf{k_c})$
    $\mathbf{z_t} \leftarrow \mathcal{H}(R)$
    $\Delta_t \leftarrow \mathbf{g}(\mathbf{x_t}, \mathbf{z_t})$
    // Compute $\mathcal{L}_{gap}, \mathcal{L}_{hist}, \mathcal{L}_{smooth}$
    $\mathcal{L}_{total} \leftarrow \texttt{computeLoss}(\Delta_t, H_t, \mathbf{k})$
    $\mathbf{k} \leftarrow \texttt{Optimize}(\mathbf{k}, \nabla_\mathbf{k}\mathcal{L}_{total})$
    $H_t \leftarrow [H_t, \mathbf{x_t} + \Delta_t]$
    $\mathbf{x_t} \leftarrow \mathbf{x_t} + \Delta_t$
```

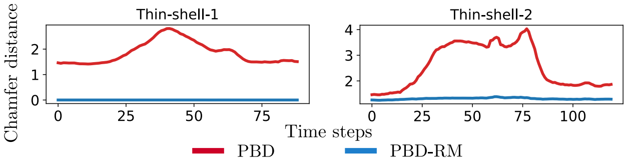

Fig. 3. Comparison of the real-to-sim Chamfer distance (smoothed) between PBD and PBD-RM. In all four trajectories, the residual mapping module significantly reduces the Chamfer distance.

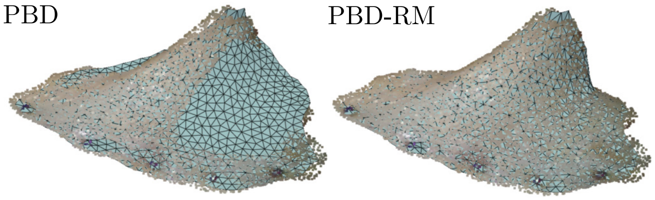

Fig. 4. Surface point cloud observations overlaid on simulation meshes. With the residual mapping, the simulation mesh matches to textured point cloud observation better than the original PBD.

## III. EXPERIMENTS & PRELIMINARY RESULTS

### A. *Real Setup & Perception Pipeline*

The proposed real-to-sim framework is evaluated in two real-world deformable object manipulation examples of thin-shell. Our experimental procedures involve the utilization of the da Vinci Research Kit (dVRK) [19], employing its robotic gripper to precisely manipulate soft tissue by executing predefined trajectories. Simultaneously, a stereo reconstruction pipeline processes stereo images captured by

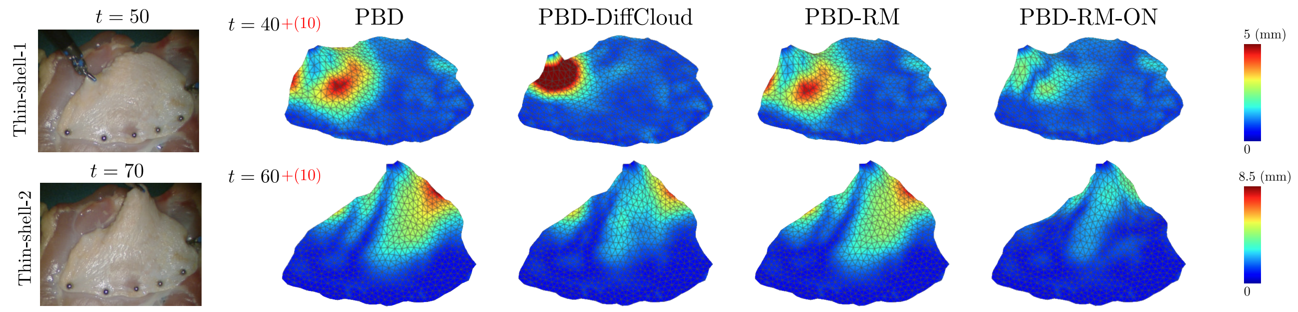

Fig. 5. Comparison of future gaps at selected time points. The gaps are visualized at $t = a + (b)$ time steps, meaning using the stiffness parameters at time $a$, forward the simulation for $b$ steps and then compute a gap. Both proposed PBD-RM and PBD-RM-ON can better predict the tissue's future deformation behavior than the original PBD simulation.

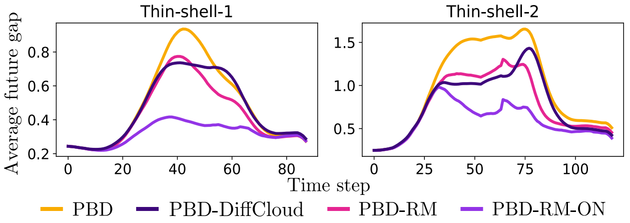

Fig. 6. Average future gap of comparison methods over time (smoothed). Oour proposed components show large error reduction over PBD.

the da Vinci endoscopic camera in 720p, producing detailed tissue surface point clouds. Figure 1 shows our physical experiment setup.

Our perception pipeline comprises stereo-depth estimation, semantic segmentation, and inverse camera projection to generate surface point cloud observations of the soft tissues. We utilize the Raft-Stereo [20] for stereo disparity estimation. Segment-Anything [21] aids in identifying image pixels corresponding to the tissue, allowing us to extract the tissue's surface point cloud from depth images. Subsequently, we employ inverse camera projection to convert the segmented depth into 3D positions. We also employ ArUco markers to determine the camera-to-world transformation. The point cloud is down-sampled to a size of 9000 points. Meshes are reconstructed from point clouds and then re-meshed to have 600 particles. The simulation's boundary conditions are selected at the locations of pins. With the above setup, we collect in total two trajectories. From now, we refer to them as Thin-shell-1, Thin-shell-2. The robotic gripper trajectories are manually labeled on the image. Later, we refer to a PBD simulation with a residual mapping module in its loop as PBD-RM and one that further performs online updates as PBD-RM-ON.

### B. *Results*

The effectiveness of applying the proposed residual mapping in the simulation loop is evaluated with Chamfer Distance. Figure 3 shows that the errors are notably reduced in both trajectories using the proposed residual mapping module. It shrinks the average errors from 1.84 mm and 2.55 mm to 0.001 mm and 1.31 mm. Figure 4 visualizes the reduced differences between point clouds and mesh with our mapping module. Our module aligns the mesh's particles while respecting PBD's geometric constraints.

We formulate an average future gap $e_t$ as our metric for evaluating the proposed online optimization strategy. Knowing the current simulation state $\mathbf{x_t}$ and a future control sequence, we can roll out a simulation, *without knowing future observations*, to get future state sequence $[\mathbf{x_{t+1}} ... \mathbf{x_{t+T}}]$. The average future gap is computed as

$$e_t = {}^{1}\!/_{T} \; \sum_{s=1}^{T} \|\mathbf{g}(\mathbf{x_{t+s}}, \mathbf{z}_{t+s})\| \tag{10}$$

For all experiments, we pick a future horizon $T = 10$. We compare to DiffCloud [15], a baseline that optimizes simulation parameters by minimizing point cloud differences on a training trajectory. we it trained with the first 30 frames of each trajectory for 25 epochs. Figure 6 visualizes the average future gaps over time. Compared to the original PBD, PBD-RM shows reduced errors, indicating that correcting the current simulation state with residual deformation leads to better future prediction. PBD-RM-ON outperforms all other method overall. The baseline produces unsatisfactory result as time goes on because it overfits to the beginning of that trajectory, whereas the PBD-RM-ON does not as it updates parameters online. Figure 5 visualizes the spatial distribution of future gaps. It indicates larger errors located where tissues are experiencing bending. Both PBD-RM and PBD-RM-ON are effective at reducing errors in those regions.

## IV. Conclusion

In this work, we presented a framework that reduces the reality to simulation gap online while performing a soft tissue manipulation task. A residual mapping module is seamlessly integrated into a simulation loop, achieving a minimal Chamfer distance between simulated particles and observation while preserving the geometric relationships in the simulator. Our optimization approach updates constraints' stiffness parameters online. In real tissue experiments, it is proven to be effective at improving predictive performance. For deploying this work into real applications, one limitation is its computation speed. Currently, PBD-RM-ON takes 0.9s to complete a step. Despite that, it can be accelerated by GPU implementation or updating the parameters asynchronously. A possible future avenue of this work would be learning residual non-linear constraints to capture more intricate tissue behaviors.

# REFERENCES

[1] Veronica E. Arriola-Rios, Puren Guler, Fanny Ficuciello, Danica Kragic, Bruno Siciliano, and Jeremy L. Wyatt. Modeling of deformable objects for robotic manipulation: A tutorial and review. *Frontiers in Robotics and AI*, 7, 2020.

[2] Hang Yin, Anastasia Varava, and Danica Kragic. Modeling, learning, perception, and control methods for deformable object manipulation. *Science Robotics*, 6(54):eabd8803, 2021.

[3] Jihong Zhu, Andrea Cherubini, Claire Dune, David Navarro-Alarcon, Farshid Alambeigi, Dmitry Berenson, Fanny Ficuciello, Kensuke Harada, Jens Kober, Xiang Li, Jia Pan, Wenzhen Yuan, and Michael Gienger. Challenges and outlook in robotic manipulation of deformable objects. *IEEE Robotics & Automation Magazine*, 29(3):67–77, 2022.

[4] Jinao Zhang, Yongmin Zhong, and Chengfan Gu. Deformable models for surgical simulation: A survey. *IEEE Reviews in Biomedical Engineering*, 11:143–164, 2018.

[5] Jack Collins, Shelvin Chand, Anthony Vanderkop, and David Howard. A review of physics simulators for robotic applications. *IEEE Access*, 9:51416–51431, 2021.

[6] Fei Liu, Entong Su, Jingpei Lu, Mingen Li, and Michael C. Yip. Robotic manipulation of deformable rope-like objects using differentiable compliant position-based dynamics. *IEEE Robotics and Automation Letters*, 8(7):3964–3971, 2023.

[7] Fei Liu, Zihan Li, Yunhai Han, Jingpei Lu, Florian Richter, and Michael C. Yip. Real-to-sim registration of deformable soft tissue with position-based dynamics for surgical robot autonomy. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12328–12334, 2021.

[8] François Faure, Christian Duriez, Hervé Delingette, Jérémie Allard, Benjamin Gilles, Stéphanie Marchesseau, Hugo Talbot, Hadrien Courtecuisse, Guillaume Bousquet, Igor Peterlik, and Stéphane Cotin. *SOFA: A Multi-Model Framework for Interactive Physical Simulation*, pages 283–321. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[9] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 737–744, 2020.

[10] Erica Salvato, Gianfranco Fenu, Eric Medvet, and Felice Andrea Pellegrino. Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning. *IEEE Access*, 9:153171–153187, 2021.

[11] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8973–8979, 2019.

[12] Jan Matas, Stephen James, and Andrew J. Davison. Sim-to-real reinforcement learning for deformable object manipulation. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 734–743. PMLR, 29–31 Oct 2018.

[13] Paul Maria Scheikl, Eleonora Tagliabue, Balázs Gyenes, Martin Wagner, Diego Dall'Alba, Paolo Fiorini, and Franziska Mathis-Ullrich. Sim-to-real transfer for visual reinforcement learning of deformable object manipulation for robot-assisted surgery. *IEEE Robotics and Automation Letters*, 8(2):560–567, 2023.

[14] Yuqing Du, Olivia Watkins, Trevor Darrell, Pieter Abbeel, and Deepak Pathak. Auto-tuned sim-to-real transfer. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1290–1296, 2021.

[15] Priya Sundaresan, Rika Antonova, and Jeannette Bohg. Diffcloud: Real-to-sim from point clouds with differentiable simulation and rendering of deformable objects, 2022.

[16] Rika Antonova, Jingyun Yang, Priya Sundaresan, Dieter Fox, Fabio Ramos, and Jeannette Bohg. A bayesian treatment of real-to-sim for deformable object manipulation. *IEEE Robotics and Automation Letters*, 7(3):5819–5826, 2022.

[17] Mehrnoosh Afshar, Tyler Meyer, Ron S. Sloboda, Siraj Hussain, Nawaid Usmani, and Mahdi Tavakoli. Registration of deformed tissue: A gnn-vae approach with data assimilation for sim-to-real transfer. *IEEE/ASME Transactions on Mechatronics*, pages 1–9, 2023.

[18] Olga Sorkine and Marc Alexa. As-Rigid-As-Possible Surface Modeling. In Alexander Belyaev and Michael Garland, editors, *Geometry Processing*. The Eurographics Association, 2007.

[19] Peter Kazanzides, Zihan Chen, Anton Deguet, Gregory S. Fischer, Russell H. Taylor, and Simon P. DiMaio. An open-source research kit for the da vinci® surgical system. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6434–6439, 2014.

[20] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *International Conference on 3D Vision (3DV)*, 2021.

[21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.